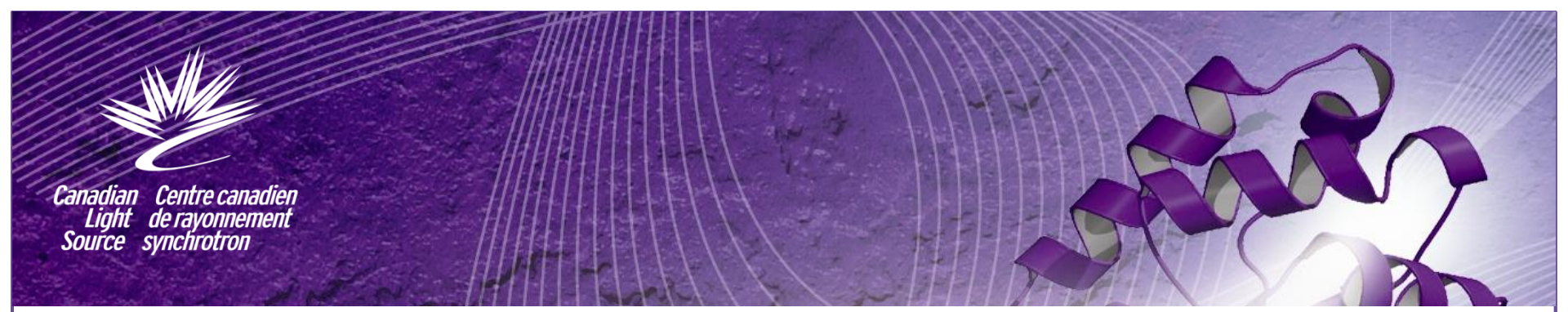
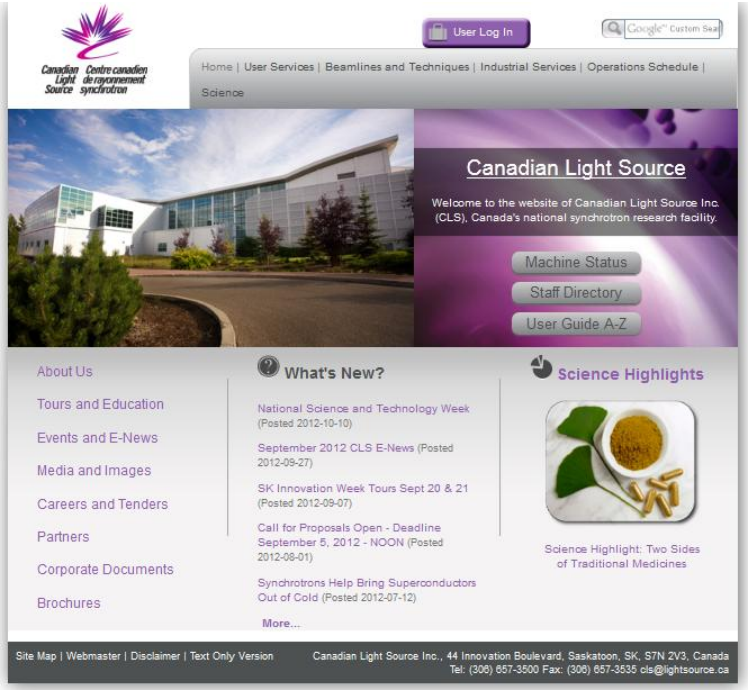


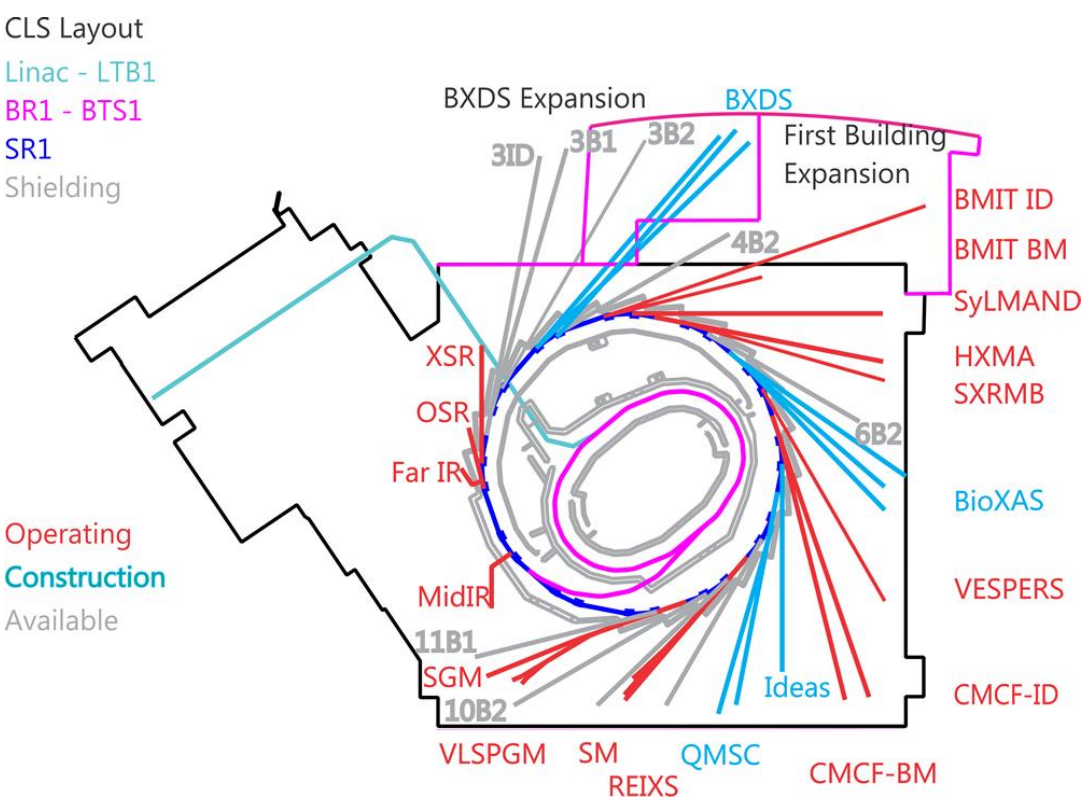
Using ezcalDL to connect to EPICS Channel Access from SHADOWVUI for Dynamic X-ray Tracing

Alan Duffy

Canadian Light Source

Introduction





www.lightsource.ca

Intro: CLS Controls Software

- Control system is based on EPICS, RTEMS using GNU GCC, Borland C++ Builder and MKS.
- PLCs are based on either MODICON Momentum or Siemens Simatic lines.
- VME Equipment is from CAEN, GE-Fanuc/VMIC, Hytec, ICS, OMS, Sensory, and WEINER.
- PC Equipment from Dell, Kontrol/PEP Modular, and Tri-M.
- Enclosures from LCH Resource, Industrial Computers, and Hammond.
- Process Instrumentation: Alltemp, Greystone, Newport, Temco Controls, Wika.
- Routine Supplies Gescan

Outline

- Motivation
- Software perquisites (what you need)
- Software description (what it does)
- Simulation model of real-life beamline
- EPICS and ezcaIDL (connections)
- ezcaSHADOWVUI (dynamic ray tracing)

Motivation

- Ray-tracing (in Shadow or ShadowVUI) is typically used during the design stage to optimize beamline performance.
- Model is static and requires user to input positions to match beamline configuration.
- After the beamline is built x-ray tracing is used less frequently.
- Automating process makes life easier.

Software Requirements

- SHADOW (Fortran and C library of subroutines)
  - Ray tracing engine developed at Nanotech Wisconsin (University of Wisconsin)
  - Used to study flashlights to x-ray telescopes and microscopes
- XOP + SHADOWVUI (written in IDL)
  - Visual User Interface for SHADOW
- EPICS with extensions: ezca, ezcaIDL
  - Provides Channel Access (CA) to process variables

SHADOW

- Main program and utilities
- I/O session driven to define system
- SHADOW Structure
  - Data files (usually binary)
  - Parameter files (e.g. START.XX in NAMELIST format)

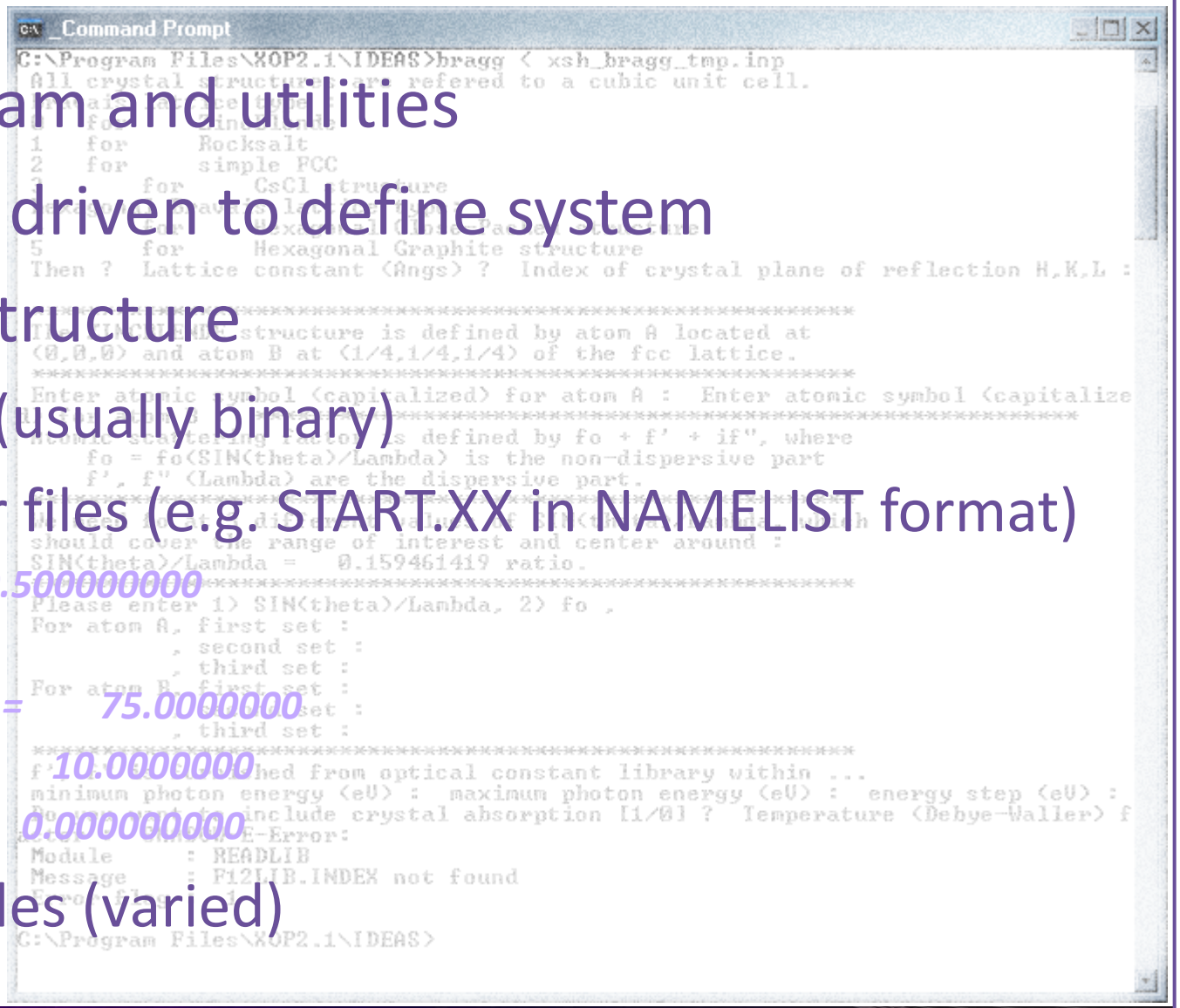
    X_ROT = -0.500000000

    ...

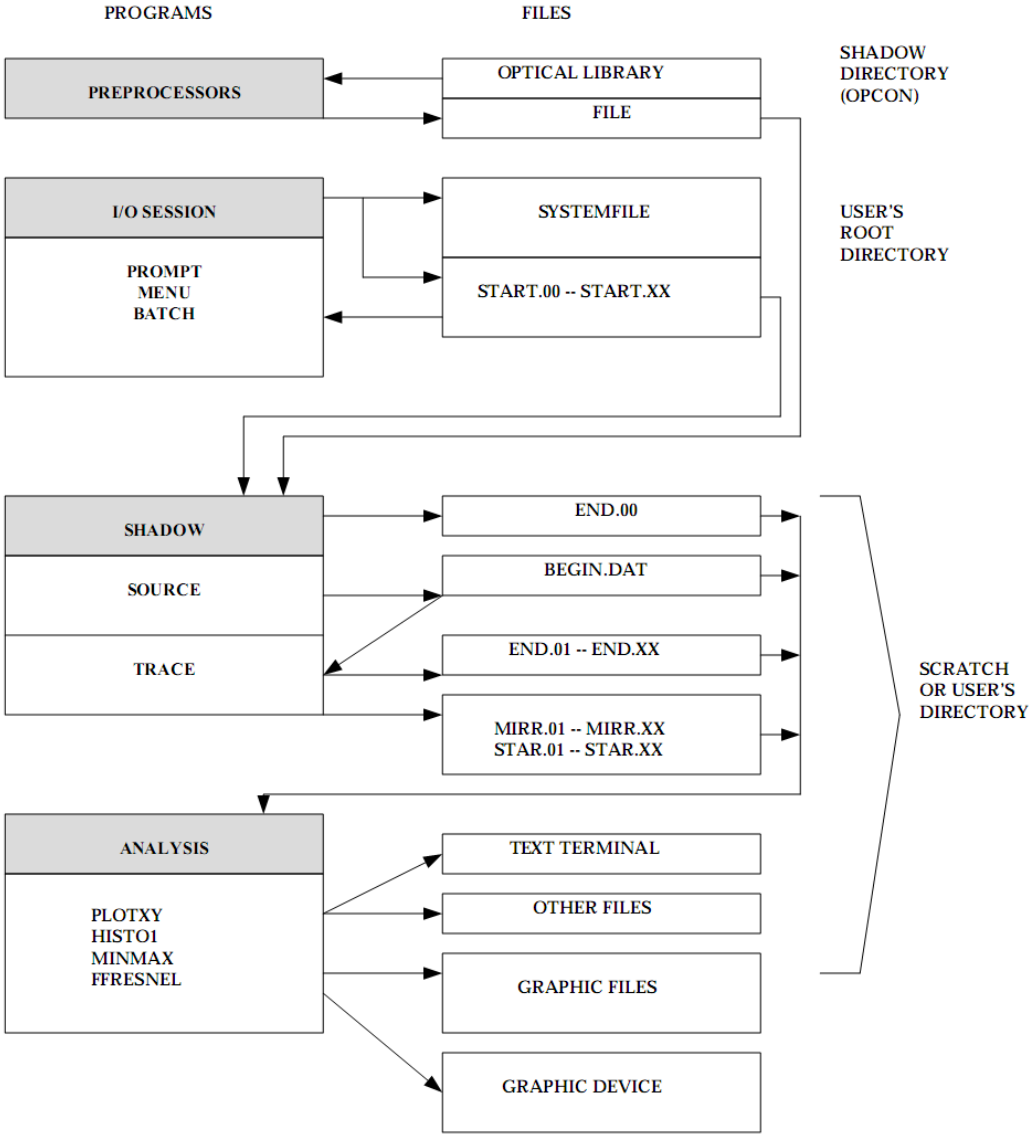
    T_INCIDENCE = 75.0000000

    T_SOURCE = 10.0000000

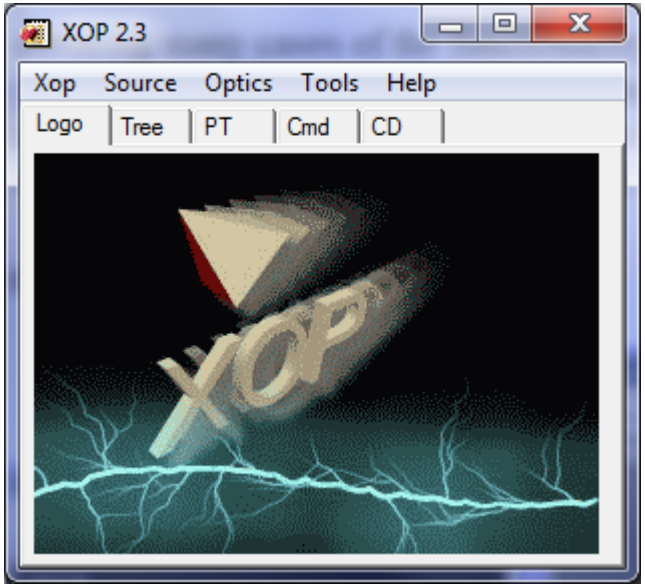
    T_IMAGE = 0.000000000
  - Analysis files (varied)

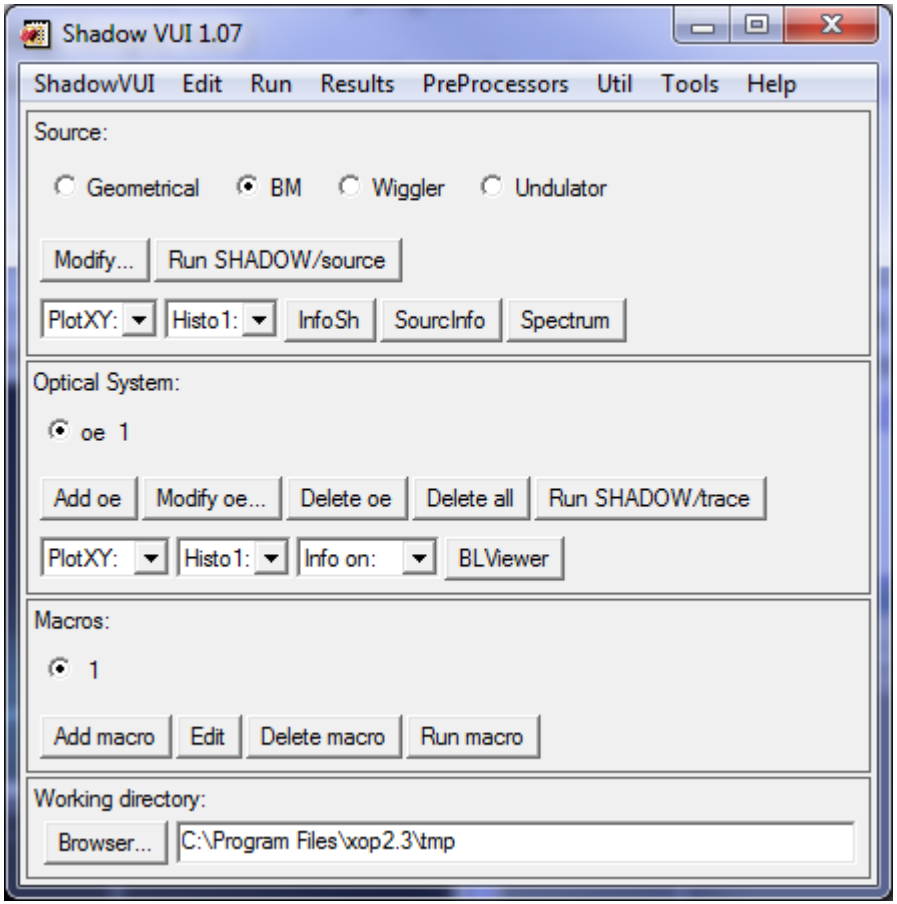


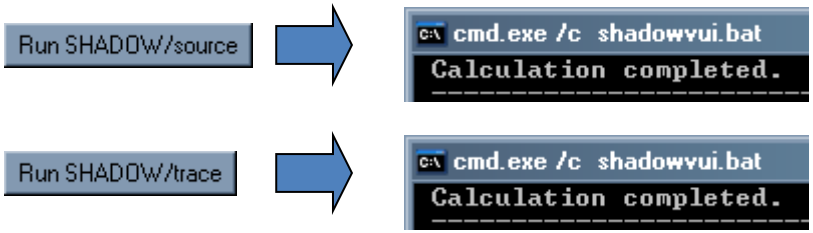
SHADOW Structure



XOP + SHADOWVUI

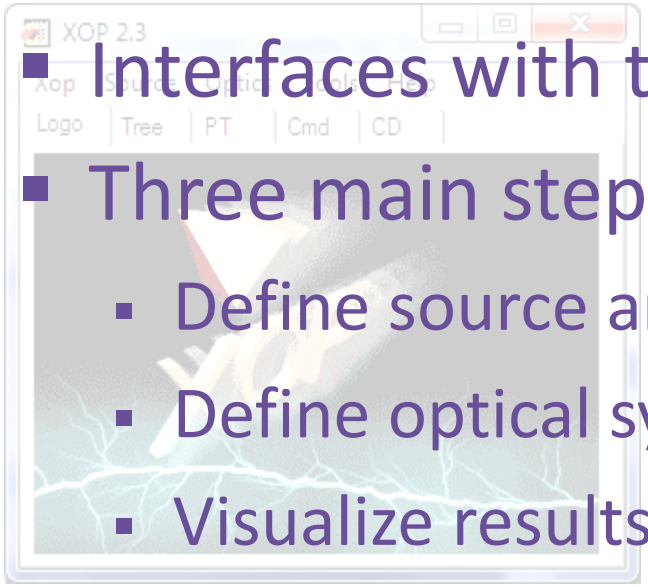


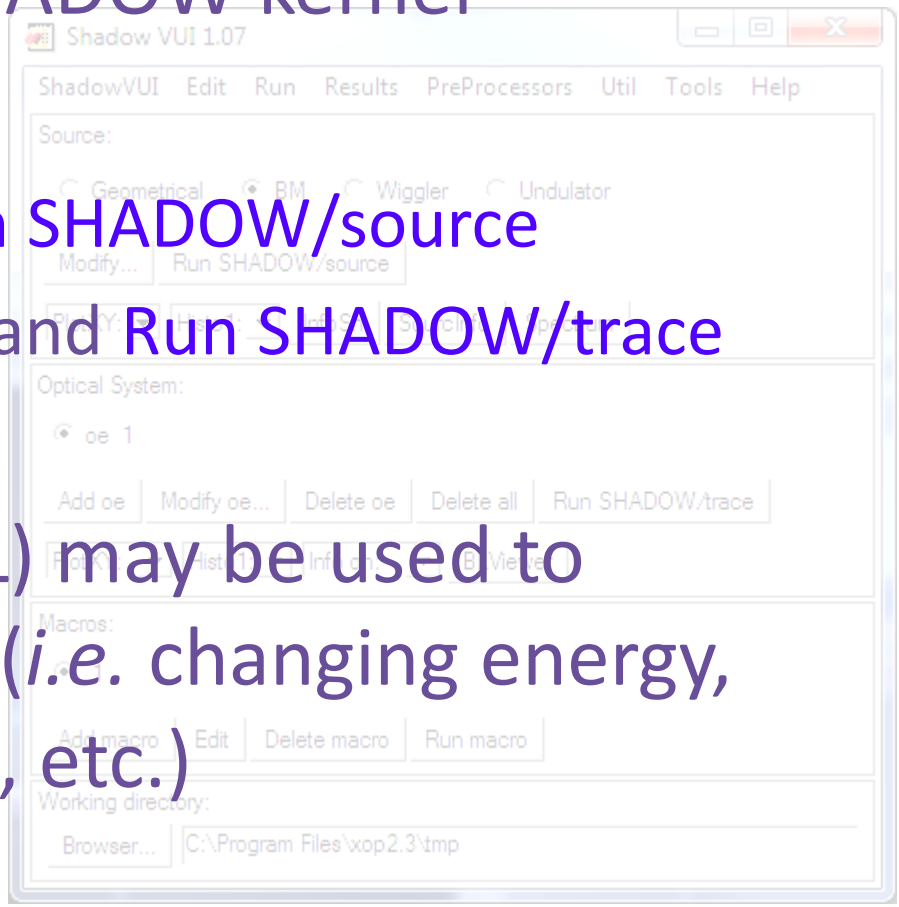


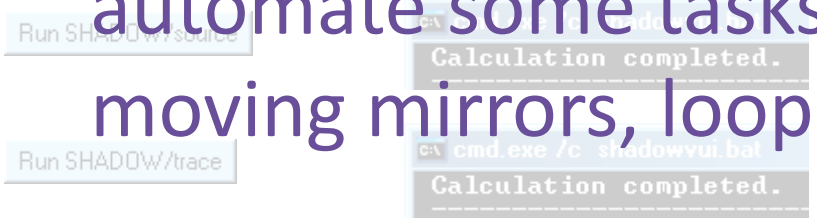


SHADOWVUI

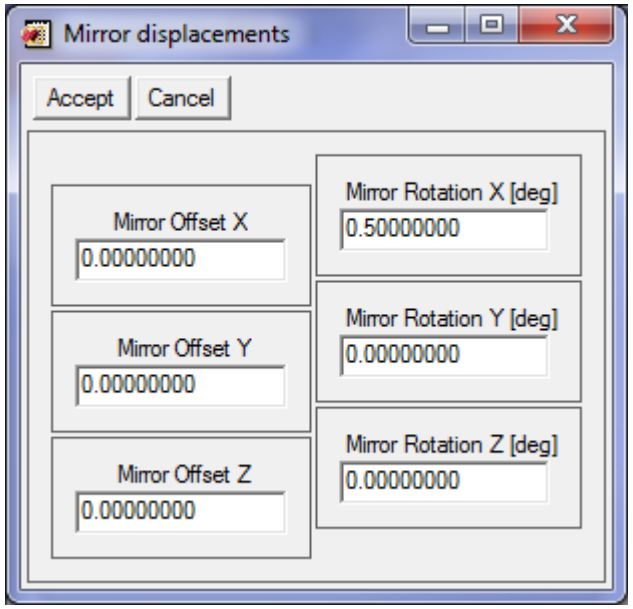
- Interfaces with the SHADOW kernel
- Three main steps:
  - Define source and Run SHADOW/source
  - Define optical system and Run SHADOW/trace
  - Visualize results
- Macros (written in IDL) may be used to automate some tasks (*i.e.* changing energy, moving mirrors, loops, etc.)





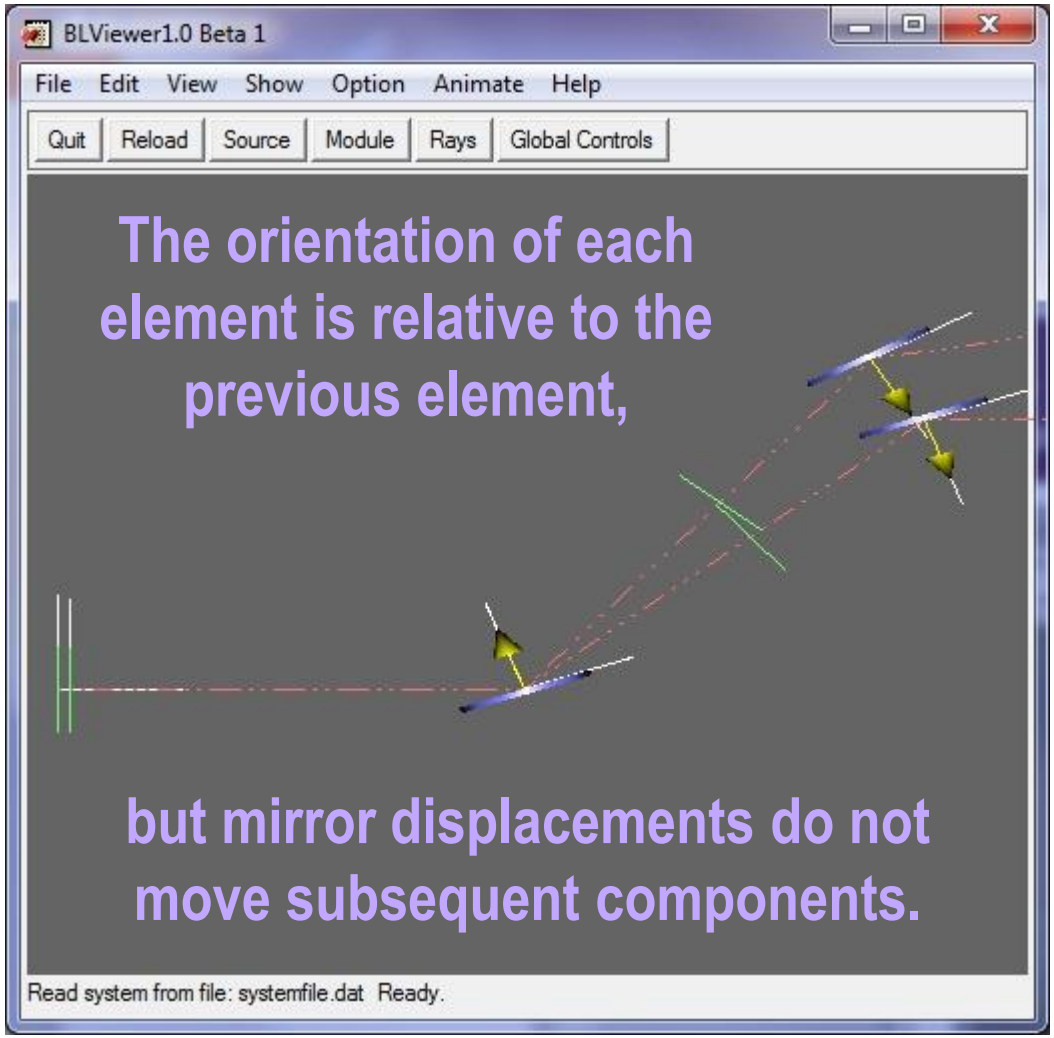


SHADOWVUI Simulation Model



- SHADOW variables

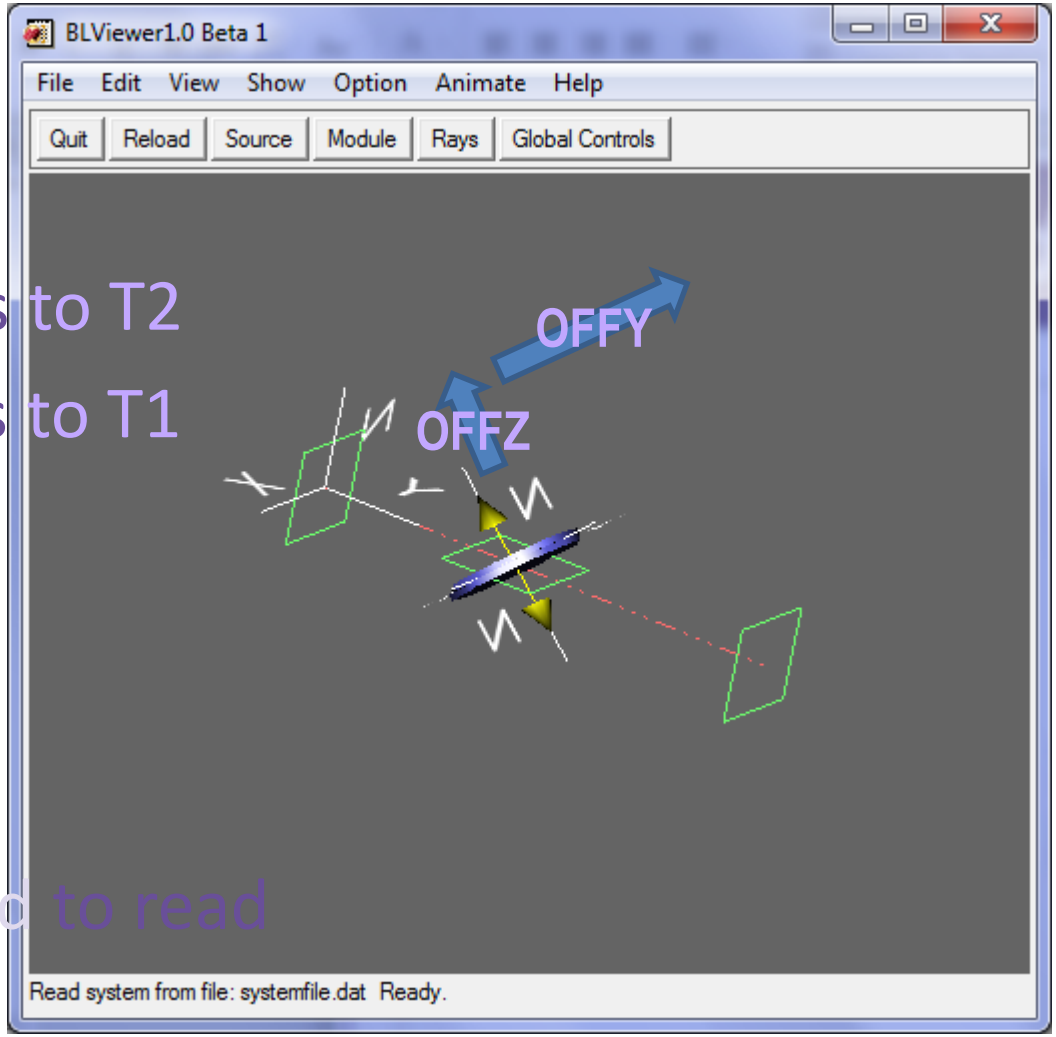
OFFX	X_ROT
OFFY	Y_ROT
OFFZ	Z_ROT



SHADOWVUI Simulation Model

- In this model
  - OFFY corresponds to T2
  - OFFZ corresponds to T1
- Time to plug and play with EPICS

This is hard to read



EPICS and ezcaIDL

- EPICS
  - real-time control system for beamlines etc.
  - process variables indicate positions of optics
- ezcaIDL
  - allows access to a set of simplified IDL interface commands to connect to Channel Access

```
Status = caGet(pvname, value, /string, max=max)
Status = caSetMonitor(pvname)
Status = caWidgetSetMonitor(name,widget_id,time=time)
```

ezcaSHADOWVUI

- Initializes ezcaIDL

```
caInit
caSetTimeout,0.001
caPendIO,time=0.01,list_time=3.
caPendEvent,time=0.000001
add_caPendEvent,timer=5.0
```

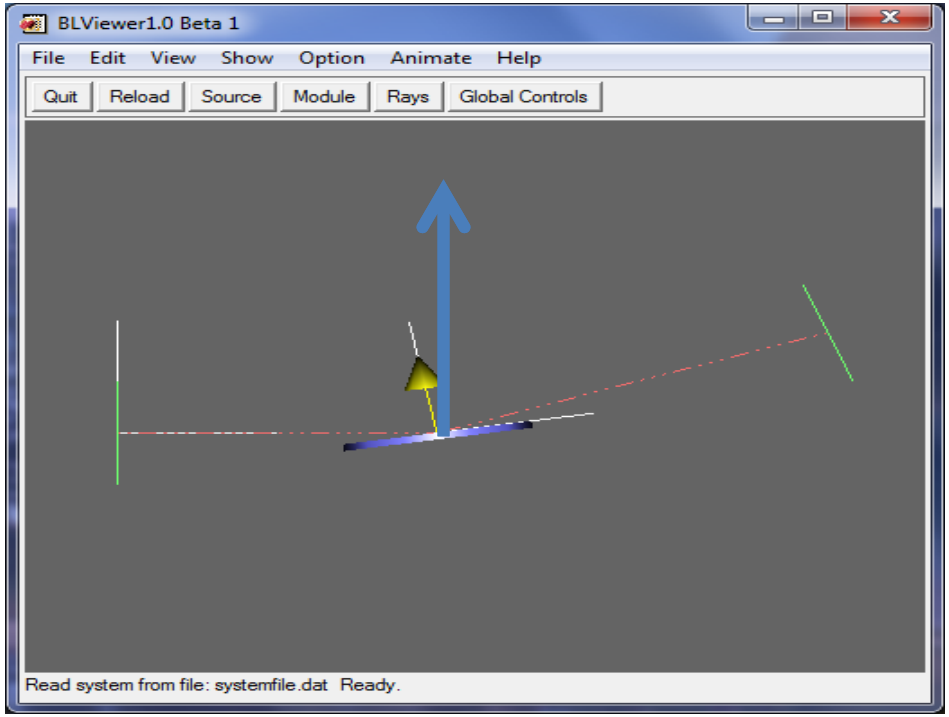
- Accesses SHADOW variables via SHADOWVUI
- Requires user input that defines relationship between model variables and beamline PVs in an IDL structure

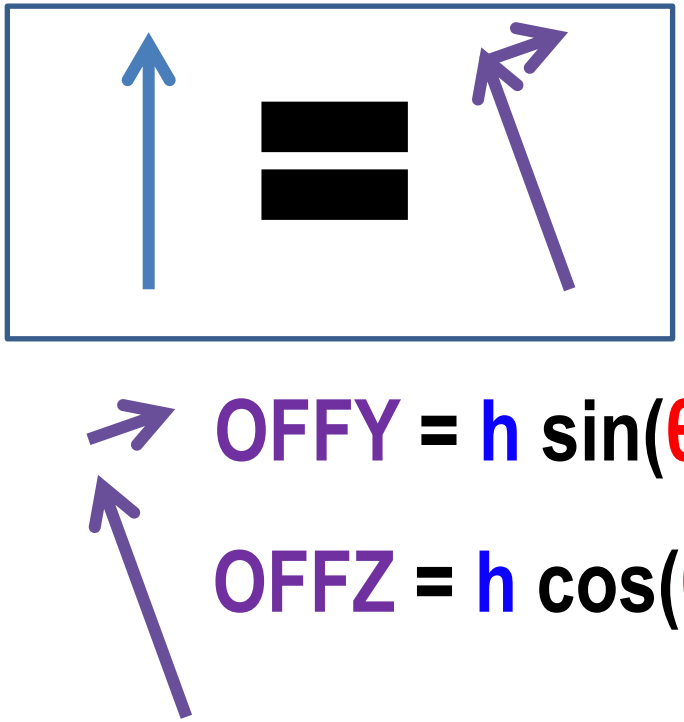
PV_INFO Structure (container)

Field	Type	Description
pv	string	EPICS process variable string
desc	string	Text to describe process variable
pv_min	float	Lower limit
pv_max	float	Upper limit
oe_num	int	Optical element number (zero otherwise)
src_num	int	Screen number (zero otherwise)
pv_2vui	string	Equation(s) to convert value of PV(s) to SHADOWVUI variable
vui_2pv	string	To convert value of SHADOWVUI variables(s) to PV value
vui_val	float	Stores SHADOWVUI variable value

- **vui_2pv string is executed on widget start-up**
- **pv_2vui string is executed on PV events**

SHADOWVUI variables and PVs



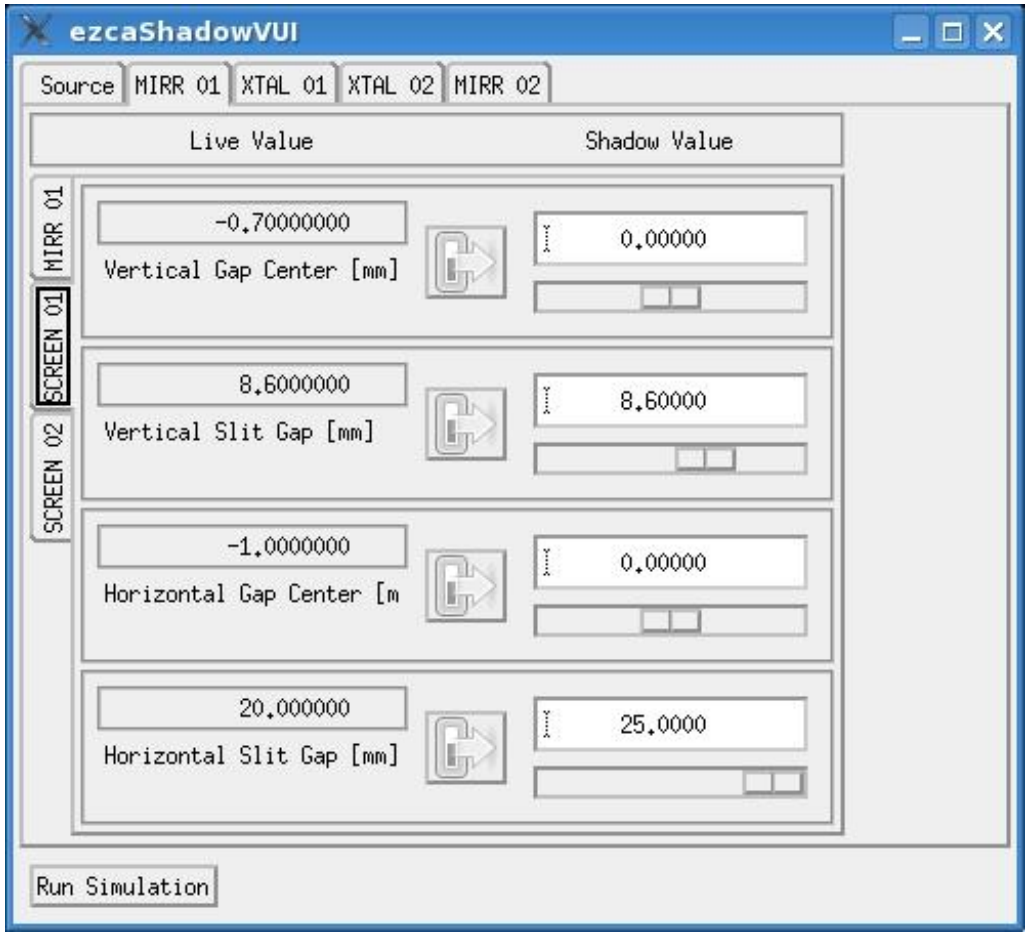


pv_2vui = '(*ptrOE1).OFFY = data_struct.h.val * sin(data_struct.theta.val)
& (*ptrOE1).OFFZ = data_struct.h.val * cos(data_struct.theta.val)'

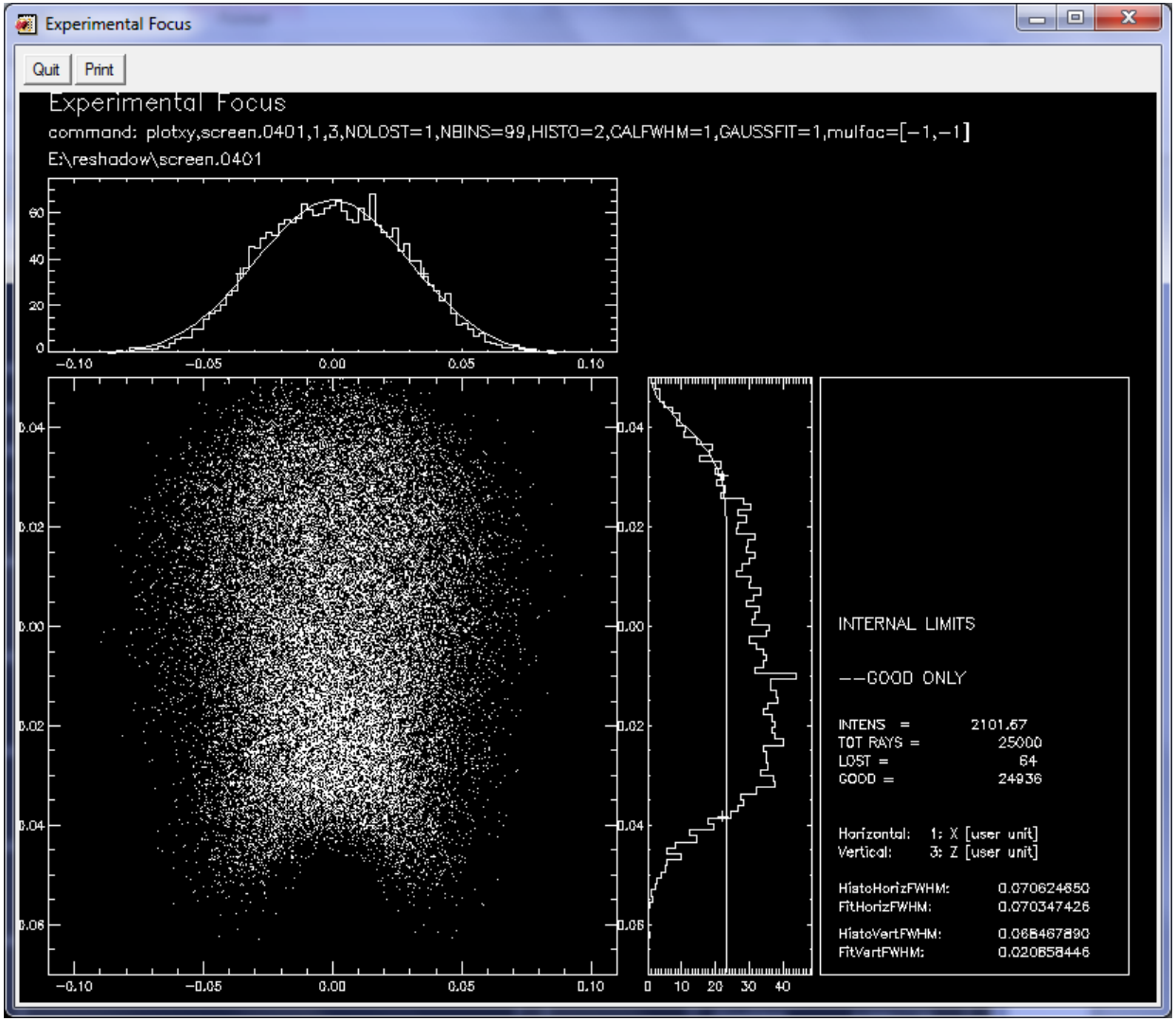
vui_2pv = 'sqrt(((*ptrOE1).OFFY)^2 +((*ptrOE1).OFFZ)^2)'

ezcaSHADOWVUI Widget

```
IDL> reshadowvui, data_struct
```



Dynamic Ray-Tracing



Summary

- SHADOW and XOP + SHADOWVUI
  - Provide ray-tracing engine and user interface
- EPICS extensions ezcaIDL/EZCA
  - allow IDL programs to access PVs
- ezcaSHADOWVUI
  - takes SHADOWVUI model and user defined relationships between PVs and model parameters
  - live positions may be used for dynamic ray tracing

Acknowledgments

Research described in this paper was performed at the **Canadian Light Source,** which is supported by:

- Natural Sciences and Engineering Research Council of Canada
- National Research Council Canada
- Canadian Institutes of Health Research
- Province of Saskatchewan
- Western Economic Diversification Canada, and
- University of Saskatchewan.

Funding Partners

Canada Foundation for Innovation
Fondation canadienne pour l'innovation

NSERC CRSNG

NRC·CNRC Canada

CIHR IRSC
Canadian Institutes of Health Research / Instituts de recherche en santé du Canada

Government of Saskatchewan

Canada
Western Economic Diversification Canada / Diversification de l'économie de l'Ouest Canada

Government of Alberta

University of Saskatchewan

Ontario

British Columbia
The Best Place on Earth

Alberta Heritage Foundation for Medical Research

Ressources naturelles Canada
Natural Resources Canada

SaskPower
Powering the future

University of Alberta

City of Saskatoon

Boehringer Ingelheim

Western

Rockwell Collins

Breast Cancer Society of Canada

RUH Foundation
Your community. Your health. Your foundation.

SHRF
Saskatchewan Health Research Foundation

Heart & Stroke Foundation of Saskatchewan
Finding answers. For life.

Saskatchewan Cancer Agency
Excellence in Patient Care and Cancer Control

Saskatoon Health Region

Regina Qu'Appelle Health Region

Alberta Cancer Foundation
25th Anniversary

Canadian Cancer Society / Société canadienne du cancer

38 supporting University Partners and growing…

Appendix - Prerequisites

- EPICS installed with extensions directory setup
  - /opt/epics/**base**
    - **base**R3.14.9.tar.gz
  - /opt/epics/**extensions**
    - **extensions**Top_20070703.tar.gz
    - **extensions**Configure_20070703.tar.gz
  - /opt/epics/extensions/src/ **(ezca,ezcaIDL,EzcaScan)**
    - **ezca**_20070625.tar.gz
    - **ezcaIDL**_20070625.tar.gz
    - **EzcaScan**_20090319.tar.gz

Install procedure (libezcaIDL.so)

- *cd /opt/epics/extensions && make*
  - ln –s /usr/local/bin/g++ /usr/bin
  - ln –s libncurses.so libcurses.so
  - yum install mingw32-readline
  - ln -s /usr/i686-pc-mingw32/sys-root/mingw/include/readline /opt/epics/base/readline
- Set environment variable EZCA_IDL_SHARE
  - /opt/epics/extensions/lib/linux-x86_64/libezcaIDL.so
- /etc/ld.so.conf.d/
  - create ezcaIDL.conf with path to libezcaIDL.so